

[illegible]

```
000000  P P P P P P P P  D D D D D D D D  R R R R R R R R  I I I I I I  V V  V V  E E E E E E E E  R R R R R R R R
000000  P P P P P P P P  D D D D D D D D  R R R R R R R R  I I I I I I  V V  V V  E E E E E E E E  R R R R R R R R
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  P P P P P P P P  DD  DD  R R R R R R R R  II  II  V V  V V  E E E E E E E E  R R R R R R R R
00 00  P P P P P P P P  DD  DD  R R R R R R R R  II  II  V V  V V  E E E E E E E E  R R R R R R R R
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
00 00  PP  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
000000  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
000000  PP  DD  DD  RR  RR  II  II  V V  V V  E E E E E E E E  RR  RR  RR  RR
```

```
LL  I I I I I I  S S S S S S S S
LL  I I I I I I  S S S S S S S S
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  S S S S S S
LL  II  S S S S S S
LL  II  SS
LL  II  SS
LL  II  SS
LL  II  SS
LLLLLLLLLL  I I I I I I  S S S S S S S S
LLLLLLLLLL  I I I I I I  S S S S S S S S
```

(5)	133	CONSOLE CONTROLLER INITIALIZATION
(6)	156	CONSOLE UNIT INITIALIZATION
(7)	194	CONSOLE RECIEVER INTERRUPT DISPATCHER
(8)	251	START I/O ON CONSOLE INTERFACE
(9)	317	CONSOLE TRANSMITTER INTERRUPT SERVICE
(9)	444	CONSOLE PORT ACTION ROUTINES
(10)	513	SEND COMMAND TO CONSOLE
(11)	553	"ALLOCATE" CONSOLE TERMINAL
(12)	592	RELEASE CONSOLE TERMINAL



```
0000 1      .TITLE OPDRIVER - VAX/VMS CONSOLE TERMINAL DRIVER
0000 2      .IDENT 'V04-000'
0000 3
0000 4 *****
0000 5
0000 6 *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7 *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8 *  ALL RIGHTS RESERVED.
0000 9
0000 10 *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11 *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12 *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13 *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14 *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15 *  TRANSFERRED.
0000 16
0000 17 *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18 *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19 *  CORPORATION.
0000 20
0000 21 *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22 *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23
0000 24 *****
0000 25
0000 26
0000 27 ++
0000 28 FACILITY:
0000 29
0000 30 VAX/VMS I/O SUBSYSTEM
0000 31
0000 32 ABSTRACT:
0000 33
0000 34
0000 35 AUTHOR: Trudy Matthews, Benn Schreiber
0000 36
0000 37 MODIFIED BY:
0000 38 V03-015 WHM0001 Bill Matthews 09-Jul-1984
0000 39 Add routines CON$PUTCHAR and CON$GETCHAR to do non-interrupt
0000 40 driven I/O to the console terminal.
0000 41
0000 42 V03-014 TCM0003 Trudy C. Matthews 22-Feb-1984
0000 43 Add routine CON$RELEASECTY which is the complement of
0000 44 CON$OWNCTY; it should be used to relinquish exclusive
0000 45 use of the console terminal.
0000 46
0000 47 V03-013 TCM0002 Trudy C. Matthews 13-Dec-1983
0000 48 Modify interface to CON$OWNCTY: it now returns the values
0000 49 that should be restored to TXCS and RXCS when the caller
0000 50 is done with "exclusive" use of the console terminal.
0000 51
0000 52 V03-012 MIR2070 Michael I. Rosenblum 04-Aug-1983
0000 53 Make reference to OPAS$VECTOR general addressing mode
0000 54
0000 55 V03-011 MIR1070 Michael I. Rosenblum 03-Aug-1983
0000 56 Add definitions for $DPTDEF.
0000 57
```

0000	58	:	V03-010	MIR0070	Michael I. Rosenblum	13-Jul-1983
0000	59	:			Remove code that was STOP2 related (the entry is obsolete)	
0000	60	:			Change code to use the CLASS_UNIT_INIT macro.	
0000	61	:				
0000	62	:	V03-009	TCM0004	Trudy C. Matthews	21-Feb-1983
0000	63	:			Add CON\$OWNCTY routine.	
0000	64	:				
0000	65	:	V03-008	MIR0027	Michael I. Rosenblum	11-Feb-1983
0000	66	:			Fix bug in new power fail code.	
0000	67	:				
0000	68	:	V03-007	MIR0022	Michael I. Rosenblum	19-Jan-1982
0000	69	:			Remove reverences to UCBSQ TT_STATE and move them into	
0000	70	:			the class driver jacket routines, to iliminate redundancy	
0000	71	:			in the port driver code.	
0000	72	:				
0000	73	:	V03-006	MIR0017	Michael I. Rosenblum	05-Jan-1982
0000	74	:			Add to unit-init routine a call back to the class driver	
0000	75	:			on powerfail. This will allow the termial driver to take	
0000	76	:			postive action on a powerfail.	
0000	77	:				
0000	78	:	V03-005	MIR0016	Michael I. Rosenblum	04-Jan-1982
0000	79	:			Change code to reflect the new port driver interface.	
0000	80	:			See DZDRIVER for detailed description of changes.	
0000	81	:				
0000	82	:	V03-004	TCM0003	Trudy C. Matthews	30-Dec-1982
0000	83	:			Fix bug in CON\$SENDCONSCMD that didn't set the select field	
0000	84	:			properly before sending the command to the console.	
0000	85	:				
0000	86	:	V03-003	TCM0002	Trudy C. Matthews	16-Dec-1982
0000	87	:			Document extra input registers to CON\$SENDCONSCMD (currently	
0000	88	:			the extra inputs registers are only used in the 11/790	
0000	89	:			version of CON\$SENDCONSCMD).	
0000	90	:				
0000	91	:	V03-002	TCM0001	Trudy C. Matthews	10-Nov-1982
0000	92	:			Change CON\$SENDCONSCMD to accept SRM-defined values as	
0000	93	:			function codes.	
0000	94	:				
0000	95	:	V03-001	KTA3018	Kerbey T. Altmann	30-Oct-1982
0000	96	:			Change psect name	
0000	97	:				
0000	98	:				

```

0000 100 :
0000 101 : SYMBOL DEFINITIONS
0000 102 :
0000 103 :
0000 104 $CRBDEF
0000 105 $CONDEF
0000 106 $DCDEF
0000 107 $DDBDEF
0000 108 $DEVDEF
0000 109 $DPTDEF
0000 110 $DYNDEF
0000 111 $IDBDEF
0000 112 $IPLDEF
0000 113 $IRPDEF
0000 114 $PRDEF
0000 115 $TTDEF
0000 116 $UCBDEF
0000 117 $TTYDEFS
0000 118 $TTYMACS
0000 119 $VECDEF
0000 120 $WCBDEF

```

```

: DEFINE CRB
: DEFINE CONSOLE FUNCTION CODES
: DEFINE DEVICE CLASSES
: DEFINE DDB
: DEFINE DEVICE CHARACTERISTICS
: DEFINE DPT
: STRUCTURE TYPE CODE DEFINITIONS
: DEFINE IDB
: DEFINE IPL LEVELS
: DEFINE IRP OFFSETS
: DEFINE PROCESSOR REGISTERS
: DEFINE TERMINAL CHARACTERISTICS
: DEFINE UCB
: TTY UCB extension (must FOLLOW $UCBDEF)
: TTY macro definitions
: DEFINE CRB VECTOR
: Define WCB

```



```
0000 122
0000 123 :
0000 124 : OUTPUT INTERRUPT QUEUE
0000 125 :
0000 126 :
00000000 127 .PSECT SYSLOA, LONG
0000 128
FF 0000 129 CURR: .BYTE -1
FF 0001 130 NEXT: .BYTE -1
00000004 0002 131 DATA: .BLKW 1
```

:CURRENT UNIT EXPECTING OUTPUT DONE  
:NEXT UNIT TO OUTPUT  
:NEXT DATA TO OUTPUT

```
0004 133 .SBTTL CONSOLE CONTROLLER INITIALIZATION
0004 134 :++
0004 135 :CONSINITIAL - INITIALIZE CONSOLE CONTROLLER
0004 136 :
0004 137 :FUNCTIONAL DESCRIPTION:
0004 138 :
0004 139 :THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE CONTROLLER.
0004 140 :
0004 141 :INPUTS:
0004 142 :
0004 143 :R5 = UCB ADDRESS
0004 144 :R9 = CRB ADDRESS
0004 145 :
0004 146 :OUTPUTS:
0004 147 :
0004 148 :ALL REGISTERS ARE PRESERVED.
0004 149 :--
0004 150 :CONSINITIAL::
20 00000040 8F DA 0004 151 MTPR #^X040,#PRS_RXCS ; INITIALIZE CONSOLE INTERFACE
22 00000040 8F DA 0008 152 MTPR #^X040,#PRS_TXCS ; ENABLE INPUT INTERRUPTS
EA AF 01 AE 0012 153 MNEGW #1,CURR ; ENABLE OUTPUT INTERRUPTS
05 0016 154 RSB ; SET NO UNITS WAITING FOR OUTPUT
```



```
0017 156 .SBTTL CONSOLE UNIT INITIALIZATION
0017 157 :++
0017 158 :CONSINITIAL - INITIALIZE CONSOLE UNIT
0017 159 :
0017 160 :FUNCTIONAL DESCRIPTION:
0017 161 :
0017 162 :THIS ROUTINE IS USED AT SYSTEM STARTUP TO INITIALIZE THE CONSOLE UNITS.
0017 163 :
0017 164 :INPUTS:
0017 165 :
0017 166 :R5 = UCB ADDRESS
0017 167 :R9 = CRB ADDRESS
0017 168 :
0017 169 :OUTPUTS:
0017 170 :
0017 171 :ALL REGISTERS ARE PRESERVED.
0017 172 :--
0017 173 :CONSINITLINE::
0017 174 :BBC #DEV$V_TRM,UCB$L_DEVCHAR(R5),CONSNULL; BR IF NOT TERMINAL
001C 175 :PUSHL R0 ; SAVE R0
001E 176 :MOVAL G^OPASVECTOR,R0 ; GET THE VECTOR ADDRESS
0025 177 :CLASS_UNIT_INIT ; AND INIT THIS UNIT
006E 178 :MOVL UCB$L TT CLASS(R5),R0 ; ADDRESS OF CLASS VECTOR TABLE
0073 179 :JSB @CLASS_SETUP_UCB(R0) ; INITIALIZE THE UCB FOR CONSOLE TERMINAL
0076 180 30$: BBC #UCB$V_POWER,UCB$W_STS(R5),40$; DID WE DETECT A POWER FAIL
007B 181 :MOVL UCB$L TT CLASS(R5),R0 ; GET THE CLASS VECTOR TABLE ADDRESS
0080 182 :JSB @CLASS_POWERFAIL(R0) ; AND GOTO THE POWERFAIL CODE
0083 183 :
0083 184 40$: POPL R0 ; RESTORE R0
0086 185 :
0086 186 :CONS$DISCONNECT::
0086 187 :CONS$INIT CTY::
0086 188 :CONS$SET CINE::
0086 189 :CONS$DS SET::
0086 190 :CONS$SET MODEM::
0086 191 :CONS$NULL:: ; NULL ENTRY FOR CONSOLE TERMINAL DISCONNECT
05 0086 192 :RSB
```

```
0087 194 .SBTTL CONSOLE RECIEVER INTERRUPT DISPATCHER
0087 195 :++
0087 196 :CONSINTINP - CONSOLE INTERRUPT ON INPUT READY
0087 197 :
0087 198 :FUNCTIONAL DESCRIPTION:
0087 199 :
0087 200 :THIS ROUTINE IS ENTERED AS A RESULT OF A RECEIVER INTERRUPT ON THE
0087 201 :CONSOLE INTERFACE. THE INTERRUPT CAN BE GENERATED BY THE CONSOLE
0087 202 :TERMINAL OR FLOPPY DRIVE.
0087 203 :
0087 204 :CONSOLE TERMINAL: ALL RECEIVED DATA CHARACTERS ARE CONSIDERED
0087 205 :UNSOLICITED AND RESULT IN AN ENTRY INTO THE
0087 206 :TERMINAL DRIVER COMMON CHARACTER BUFFERING
0087 207 :ROUTINE '@UCBSL_TT_PUTNXT(R5)'.
0087 208 :
0087 209 :CONSOLE FLOPPY: RECEIVER INTERRUPTS FROM THE CONSOLE FLOPPY
0087 210 :MUST BE EXPECTED (UCBSV INT = 1). IF THE
0087 211 :INTERRUPT IS EXPECTED THEN UCBSV INTTYPE IS SET
0087 212 :IN UCBSW STS TO SIGNAL A RECEIVER INTERRUPT.
0087 213 :THE DRIVER IS THEN ENTERED AT ITS WAIT FOR INTERRUPT
0087 214 :CONTEXT PC.
0087 215 :
0087 216 :INPUTS:
0087 217 :
0087 218 :R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
0087 219 :
0087 220 :OO(SP) = ADDRESS OF THE IDB
0087 221 :
0087 222 :OUTPUTS:
0087 223 :
0087 224 :THE SAVED REGISTERS ARE RESTORED BEFORE REI.
0087 225 :--
0087 226 :CONSINTINP::
53 21 DB 0087 227 MFPR #PRS_RXDB,R3 ; MOVE DATA FROM INTERFACE
008A 228 :
008A 229 :GET THE ASSOCIATED UCB
008A 230 :
008A 231 :MOVL @ (SP)+,R4 ; GET IDB ADDRESS
52 53 54 9E DO 008D 232 ASHL #-8,R3,R2 ; GET LINE NUMBER
52 53 F8 8F 78 008D 233 BICW #^C<3>,R2 ;
52 53 FFFC 8F AA 0092 234 CMPW R2,IDBSW_UNITS(R4) ; LEGAL UNIT NUMBER?
OC A4 52 B1 0097 235 BGEQU 30$ ; DISMISS INTERRUPT
55 18 A442 D0 009D 236 5$: MOVL IDBSL_UCBLST(R4)[R2],R5 ; GET THE UCB ADDRESS
64 A5 0080 8F A8 00A4 237 BEQL 30$ ; NO UCB - DISMISS INTERRUPT
03 38 A5 E0 00AA 238 BISW #UCBSM_INTTYPE,UCBSW_STS(R5); SET RECEIVER INTERRUPT
0086 31 00AF 239 BBS #DEV$V_TRM,- ; ENTER FLOPPY INTERRUPT CODE
0082 240 UCBSL_DEVCHAR(R5),10$
0082 241 BRW FLOPINT ; IF NOT TERMINAL UCB
0082 242 :
0082 243 :CONSOLE TERMINAL INTERRUPT
53 53 9A 00B2 244 10$: MOVZBL R3,R3 ; ZERO TOP 3 BYTES
0110 D5 16 00B5 245 JSB @UCBSL_TT_PUTNXT(R5) ; BUFFER THE CHARACTER
02 13 00B9 246 BEQL 30$ ; IF EQL THEN NO CHARACTER TO OUTPUT
03 10 00BB 247 BSBB CON$STARTIO ; OUTPUT THE CHARACTER
0092 31 00BD 248 20$:
0087 249 30$: BRW DISMIS ; GO
```

```
00C0 251 .SBTTL START I/O ON CONSOLE INTERFACE
00C0 252
00C0 253 ++
00C0 254 CONSSTARTIO - START I/O ON CONSOLE INTERFACE
00C0 255
00C0 256 FUNCTIONAL DESCRIPTION:
00C0 257
00C0 258 THIS ROUTINE IS ENTERED TO OUTPUT A CHARACTER TO THE CONSOLE INTERFACE.
00C0 259 IF THE INTERFACE IS READY THE DATA IS OUTPUT DIRECTLY. IF THE INTERFACE
00C0 260 IS NOT READY THEN THE DATA IS QUEUED AND SUBSEQUENTLY OUTPUT ON THE
00C0 261 NEXT READY INTERRUPT.
00C0 262
00C0 263 IN EITHER CASE, A RETURN TO THE CALLER IS DONE TO ENTER A 'WAIT FOR
00C0 264 INTERRUPT' STATE.
00C0 265
00C0 266 INPUTS:
00C0 267
00C0 268 R3 = DATA TO OUTPUT
00C0 269 R5 = UCB ADDRESS
00C0 270
00C0 271 OUTPUTS:
00C0 272
00C0 273 R3,R4,R5 ARE PRESERVED.
00C0 274
00C0 275 .ENABLE LSB
00C0 276
00C0 277 CONSSTARTIO::
00C0 278
00C0 279 BBC #DEV$V_TRM,UCB$L_DEVCHAR(R5),5$ : BRANCH IF FLOPPY
00C5 280 BLSS 20$ : BRANCH IF BURST MODE
00C7 281
00C7 282 : INSERT UNIT NUMBER IN DATA WORD
00C7 283
00C7 284 5$:
00C7 285 CONSSTARTIO1:
00C7 286 MOVZBL UCB$W_UNIT(R5),R2 : PICK UP UNIT NUMBER
53 02 52 54 A5 9A 00CB 287 INSV R2,#8,#2,R3 : INSERT UNIT NUMBER AND CLEAR OTHER BITS
00D0 288
00D0 289 : INITIATE I/O IF NO PREVIOUS UNIT WAITING
00D0 290
00D0 291 TSTB CURR : OTHER UNIT WAITING?
00D4 292 BLSS 10$ : IF LSS NO, OUTPUT DATA AND RETURN
FF26 CF 52 90 00D6 293 MOVB R2,NEXT : SAVE UNIT NUMBER
FF22 CF 53 80 00DB 294 MOVW R3,DATA : SAVE DATA
00E0 295 RSB : EXIT
00E1 296 10$:
00E1 297 MOVB R2,CURR : SAVE UNIT NUMBER OF CURRENT SENDER
FF1A CF 52 90 00E6 298 MTPR R3,#PR$_TXDB : SEND DATA
23 53 DA 00E9 299 RSB
00EA 300
00EA 301 20$: BISW #TTY$M_TANK_BURST,- : SET BURST MODE
0108 C5 00EE 302 UCB$W_TT_HOCD(R5)
00F1 303
00F1 304 : TAKE CHARACTER OUT OF BURST BUFFER AND TRY TO OUTPUT IT IMMEDIATELY
00F1 305
53 011C D5 9A 00F1 306 MOVZBL @UCB$L_TT_OUTADR(R5),R3 : OUTPUT NEXT BYTE
011C C5 D6 00F6 307 INCL UCB$L_TT_OUTADR(R5) : UPDATE POINTER
```



0120	C5	B7	00FA	308	DECW	UCBSW_TT_OUTLEN(R5)	:	UPDATE COUNT
	C7	12	00FE	309	BNEQ	SS	:	NOT LAST CHARACTER
0800	8F	AA	0100	310	BICW	#TTY\$M_TANK_BURST,-	:	RESET BURST ACTIVE
0108	C5		0104	311		UCBSW_TT_HOED(R5)		
	BE	11	0107	312	BRB	SS		
			0109	313				
			0109	314	.DISABLE	LSB		
			0109	315				

```
0109 317 .SBTTL CONSOLE TRANSMITTER INTERRUPT SERVICE
0109 318 ++
0109 319 CON$INTOUT - CONSOLE TRANSMITTER INTERRUPT SERVICE
0109 320
0109 321 FUNCTIONAL DESCRIPTION:
0109 322
0109 323 THIS ROUTINE IS ENTERED WHEN A CONSOLE UNIT READY INTERRUPT OCCURS.
0109 324
0109 325 A SEARCH IS MADE OF THE COLD STORAGE BUFFER FOR DATA TO OUTPUT.
0109 326 IF ANY IS FOUND THEN IT IS OUTPUT. IF NO VALID DATA IS FOUND THEN
0109 327 THE INTERRUPT STATE OF THE UNIT IS CHECKED FOR EXPECTED INTERRUPT.
0109 328 IF NO INTERRUPT IS EXPECTED, THE INTERRUPT IS DISMISSED. IF AN INTERRUPT
0109 329 IS EXPECTED THEN THE DRIVER IS ENTERED. IN THE CASE OF THE CONSOLE TERMINAL,
0109 330 A SPECIFIC ROUTINE IS ENTERED TO GET THE NEXT CHARACTER AVAILABLE TO OUTPUT
0109 331 ON THE UNIT. IN THE CASE OF THE CONSOLE FLOPPY, THE WAIT FOR INTERRUPT
0109 332 CONTEXT IS TAKEN FROM THE UCB AND THE SPECIFIED ROUTINE IS ENTERED.
0109 333
0109 334 INPUTS:
0109 335
0109 336 R0,R1,R2,R3,R4,R5 ARE SAVED ON THE INTERRUPT STACK.
0109 337
0109 338 OO(SP) = ADDRESS OF THE IDB
0109 339
0109 340 OUTPUTS:
0109 341
0109 342 THE SAVED REGISTERS ARE RESTORED BEFORE REI.
0109 343
0109 344 --
0109 345 CON$INTOUT::
0109 346 MOVL @ (SP)+,R4 ; GET ADDRESS OF IDB
0109 347 CVTBL NEXT,R2 ; GET UNIT NUMBER OF NEXT SENDER
0109 348 BLSS 10$ ; IF LSS NO SENDER WAITING
0109 349 MNEGB #1,NEXT ; RESET SENDER-WAITING
0109 350 MOVZWL DATA,R3 ; GET DATA TO SEND
0109 351 MTPR R3,#PRS_TXDB ; TRANSMIT DATA
0109 352 10$:
0109 353 MOVZBL CURR,R3 ; GET UNIT NUMBER OF CURRENT SENDER
0109 354 MOVB R2,CURR ; SET UNIT NUMBER OF NEXT CURRENT
0109 355 TSTB R3 ; SPURIOUS INTERRUPT?
0109 356 BLSS DISMIS ; IF LSS YES
0109 357 MOVL IDB$U_UCBLST(R4)[R3],R5 ; GET ADDRESS OF UCB
0109 358 BBCC #UCB$V_INTTYPE,UCB$W_STS(R5),FLOPINT ; SET OUTPUT INTERRUPT
0109 359 FLOPINT:
0109 360 BBS #DEV$V_TRM,UCB$U_DEVCHAR(R5),20$ ; IF BIT CLEAR, UNIT = FLOPPY
0109 361 BBCC #UCB$V_TIM,UCB$W_STS(R5),10$ ; CLEAR TIMEOUT PENDING
0109 362 10$: BBCC #UCB$V_INT,UCB$W_STS(R5),DISMIS ; EXIT IF NOT WAITING FOR INTERRUPT
0109 363 ;
0109 364 ; DEVICE IS A FLOPPY
0109 365 ;
0109 366 MOVL UCB$U_FR4(R5),R4 ; RESTORE R4 CONTEXT
0109 367 JSB @UCB$U_FPC(R5) ; ENTER FLOPPY INTERRUPT HANDLER
0109 368 BRB DISMIS
0109 369 ;
0109 370 ; DEVICE IS A TERMINAL
0109 371 20$: BSBB TERMINALIO
0109 372
0109 373 DISMIS:
0109 374 ;
```

52	54	9E	D0	0109	345	MOVL	@(SP)+,R4	;	GET ADDRESS OF IDB	
	FEF1	CF	98	010C	346	CVTBL	NEXT,R2	;	GET UNIT NUMBER OF NEXT SENDER	
		0D	19	0111	347	BLSS	10\$	;	IF LSS NO SENDER WAITING	
FEE9	CF	01	8E	0113	348	MNEGB	#1,NEXT	;	RESET SENDER-WAITING	
53	FEE6	CF	3C	0118	349	MOVZWL	DATA,R3	;	GET DATA TO SEND	
	23	53	DA	011D	350	MTPR	R3,#PRS_TXDB	;	TRANSMIT DATA	
				0120	351	10\$:				
53	FEDC	CF	9A	0120	352	MOVZBL	CURR,R3	;	GET UNIT NUMBER OF CURRENT SENDER	
FED6	CF	52	90	0125	353	MOVB	R2,CURR	;	SET UNIT NUMBER OF NEXT CURRENT	
		53	95	012A	354	TSTB	R3	;	SPURIOUS INTERRUPT?	
		24	19	012C	355	BLSS	DISMIS	;	IF LSS YES	
55	18	A443	D0	012E	356	MOVL	IDB\$U_UCBLST(R4)[R3],R5	;	GET ADDRESS OF UCB	
00 64	A5	07	E5	0133	357	BBCC	#UCB\$V_INTTYPE,UCB\$W_STS(R5),FLOPINT	;	SET OUTPUT INTERRUPT	
				0138	358	FLOPINT:				
13 38	A5	02	E0	0138	359	BBS	#DEV\$V_TRM,UCB\$U_DEVCHAR(R5),20\$	;	IF BIT CLEAR, UNIT = FLOPPY	
00 64	A5	00	E5	013D	360	BBCC	#UCB\$V_TIM,UCB\$W_STS(R5),10\$	;	CLEAR TIMEOUT PENDING	
08 64	A5	01	E5	0142	361	10\$:	BBCC	#UCB\$V_INT,UCB\$W_STS(R5),DISMIS	;	EXIT IF NOT WAITING FOR INTERRUPT
				0147	362	;				
				0147	363	;	DEVICE IS A FLOPPY			
				0147	364	;				
54	14	A5	D0	0147	365	MOVL	UCB\$U_FR4(R5),R4	;	RESTORE R4 CONTEXT	
	0C	B5	16	0148	366	JSB	@UCB\$U_FPC(R5)	;	ENTER FLOPPY INTERRUPT HANDLER	
		02	11	014E	367	BRB	DISMIS			
				0150	368	;				
				0150	369	;	DEVICE IS A TERMINAL			
				0150	370	;				
	0A	10		0150	371	20\$:	BSBB	TERMINALIO		
				0152	372					
				0152	373	DISMIS:		;		

OPDRIVER  
V04-000

J 6  
- VAX/VMS CONSOLE TERMINAL DRIVER  
CONSOLE TRANSMITTER INTERRUPT SERVICE

16-SEP-1984 00:16:57 VAX/VMS Macro V04-00  
5-SEP-1984 04:11:02 [SYSLOA.SRC]OPDRIVER.MAR;1

Page 11  
(9)

50	BE	7D	0152	374	MOVQ	(SP)+,R0	:	RESTORE REGISTERS
52	BE	7D	0155	375	MOVQ	(SP)+,R2	:	
54	BE	7D	0158	376	MOVQ	(SP)+,R4	:	
		02	015B	377	REI		:	

OP  
VO



```
015C 379 TERMINAL IO:
015C 380 :
015C 381 : CHECK FOR BURST ACTIVE ON LINE
015C 382 :
015C 383 : CMPB #TTY$M_TANK_BURST-8,- ; ONLY BURST ACTIVE?
015E 384 : UCBSW_TT_HOLD+1(R5)
0161 385 : BEQL BURST- ; YES, CONTINUE BURST
0163 386 :
0163 387 : LOOK FOR NEXT OUTPUT STATE IN TANK
0163 388 :
0163 389 :
0163 390 : FFS #0,#6,UCBSW_TT_HOLD+1(R5),R3
016A 391 : CASE R3,TYPE=B,<= ; DISPATCH
016A 392 : PREMT,- ; send premt character
016A 393 : STOP,- ; STOP OUTPUT
016A 394 : BURST,- ; BURST IN PROGRESS
016A 395 : >
0174 396 :
0174 397 : NO PENDING DATA - LOOK FOR NEXT CHARACTER
0174 398 :
0174 399 : BICB #UCBSM_TIM!UCBSM_INT,UCBSW_STS(R5); CLEAR TIMEOUT AND EXPECTED
0178 400 :
0178 401 : CALL TERMINAL DRIVER ROUTINE
0178 402 :
0178 403 : JSB @UCBSL_TT_GETNXT(R5) ; GET THE NEXT CHARACTER
017C 404 : BLSS START_BURST ; BURST SPECIFIED
017E 405 : BEQL EXIT ; NONE
FF44 30 0180 406 : BSBW CONSTARTIO1 ; INITIATE I/O
37 11 0183 407 : BRB EXIT_INT ; EXIT WITH INTERRUPT EXPECTED
0185 408 :
0185 409 : START_BURST:
0185 410 : BISM #TTY$M_TANK_BURST,- ; SIGNAL BURST ACTIVE
0189 411 : UCBSW_TT_HOLD(R5)
018C 412 : BURST:
018C 413 : MOVZBL @UCBSL_TT_OUTADR(R5),R3 ; SET NEXT OUTPUT CHARACTER
0191 414 : INCL UCBSL_TT_OUTADR(R5) ; UPDATE POINTER
0195 415 : DECB UCBSW_TT_OUTLEN(R5) ; UPDATE COUNT
0199 416 : BNEQ 10$ ; NOT LAST CHARACTER
019B 417 : BICW #TTY$M_TANK_BURST,- ; RESET BURST ACTIVE
019F 418 : UCBSW_TT_HOLD(R5)
01A2 419 :
FF22 30 01A2 420 10$: BSBW CONSTARTIO1 ; OUTPUT CHARACTER
15 11 01A5 421 : BRB EXIT_INT ; EXIT WITH INTERRUPT EXPECTED
01A7 422 :
01A7 423 : STOP:
01A7 424 : BICB #UCBSM_INT!UCBSM_TIM,-
01A9 425 : UCBSW_STS(R5) ; RESET OUTPUT ACTIVE
01AB 426 : BRB EXIT
01AD 427 :
01AD 428 : .ENABLE LSB
01AD 429 : PREMT:
01AD 430 : BICW #TTY$M_TANK_PREMT,- ; RESET XOFF STATE
01B1 431 : UCBSW_TT_HOLD(R5)
01B4 432 : MOVZBL UCBSB_TT_PREMT(R5),R3 ; SEND premt character
01B9 433 :
01B9 434 : BSBW CONSTARTIO1 ; OUTPUT CHARACTER
01BC 435 :
```

OPDRIVER  
V04-000

- VAX/VMS CONSOLE TERMINAL DRIVER L 6  
CONSOLE TRANSMITTER INTERRUPT SERVICE

16-SEP-1984 00:16:57 VAX/VMS Macro V04-00  
5-SEP-1984 04:11:02 [SYSLOA.SRC]OPDRIVER.MAR;1

Page 13  
(9)

```
64 A5 02 88 01BC 436 EXIT_INT:
           01BC 437
           01C0 438 EXIT:
05        01C0 439
           01C1 440
           01C1 441 .DISABLE
           01C1 442 LSB
```

OP  
VO

```
01C1 444 .SBTTL CONSOLE PORT ACTION ROUTINES
01C1 445 :++
01C1 446 :CONS$OFF - SEND XOFF
01C1 447 :CONS$XON - SEND XON
01C1 448 :CONS$STOP - STOP OUTPUT
01C1 449 :CONS$STOP2 - ALTERNATE STOP
01C1 450 :CONS$ABORT - ABORT CURRENT OUTPUT
01C1 451 :CONS$RESUME - RESUME STOPPED OUTPUT
01C1 452 :
01C1 453 :FUNCTIONAL DESCRIPTION:
01C1 454 :
01C1 455 :THESE ROUTINES ARE USED BY THE THE TERMINAL CLASS DRIVER TO
01C1 456 :CONTROL OUTPUT ON THE PORT
01C1 457 :
01C1 458 :INPUTS:
01C1 459 :
01C1 460 :R5 = UCB ADDRESS
01C1 461 :
01C1 462 :OUTPUTS:
01C1 463 :
01C1 464 :R5 = UCB ADDRESS
01C1 465 :--
01C1 466 :.ENABLE LSB
01C1 467 :
01C1 468 :CONS$OFF::
01C1 469 :CONS$XON::
0108 C5 0100 8F A8 01C1 470 BISH #TTY$M_TANK_PREMPT,UCB$W_TT_HOLD(R5) : SCHEDULE PREMPT
010A C5 53 90 01C8 471 MOVB R3,UCB$B_TT_PREMPT(R5) : SAVE PREMPT CHARACTER
0C 64 A5 01 E0 01CD 472 BBS #UCB$V_INT,UCB$W_STS(R5),10$ : IF OUTPUT ACTIVE,
1E BB 01D2 473 :
FF85 30 01D2 474 PUSHR #^M<R1,R2,R3,R4> : SAVE REGISTERS
1E BA 01D4 475 BSBW TERMINALIO : START OUTPUT
00 64 A5 01 E2 01D7 476 POPR #^M<R1,R2,R3,R4> : RESTORE REGISTERS
01DE 477 BBSS #UCB$V_INT,UCB$W_STS(R5),10$ : SHOW OUTPUT ACTIVE
05 01DE 478 10$:
01DE 479 RSB
01DF 480 .DISABLE LSB
01DF 481 :
01DF 482 :CONS$STOP::
0200 8F A8 01DF 483 BISH #TTY$M_TANK_STOP,- : SCHEDULE STOP
0108 C5 05 01E3 484 UCBSW_TT_HOLD(R5)
01E6 485 RSB
01E7 486 :
0108 C5 0B E5 01E7 487 :CONS$ABORT::
01E7 488 BBCC #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),- : RESET BURST ACTIVE
00 01EC 489 10$
01ED 490 :
01ED 491 10$: TIMSET 1 : SETUP A TIMER
05 0200 492 RSB
0201 493 :
0201 494 :CONS$RESUME::
0108 C5 0200 51 DD 0201 495 PUSHL R1 : SAVE A REGISTER
2B 0108 C5 0B E1 0203 496 BICW #TTY$M_TANK_STOP- : RESET STOP CONDITIONS
51 0120 C5 3C 020A 497 UCBSW_TT_HOLD(R5) : BRANCH IF NO BURST IN PROG
0210 498 BBC #TTY$V_TANK_BURST,UCB$W_TT_HOLD(R5),40$
0210 499 :
0210 500 MOVZWL UCBSW_TT_OUTLEN(R5),R1 : NUMBER OF BURST CHARS
```



07 64 A5	01	E0	0215	501	TIMSET	R1,R1		; SET TIME OUT
	1E	BB	022F	502				
	FF23	30	022F	503	30\$:			
	1E	BA	022F	504	BBS	#UCBSV_INT,UCBSW_STS(R5),40\$		; SKIP IF OUTPUT ON
			0234	505	PUSHR	#^M<R1,R2,R3,R4>		; SAVE REGISTERS
			0236	506	BSBW	TERMINALIO		; START I/O
			0239	507	POPR	#^M<R1,R2,R3,R4>		; RESTORE REGISTERS
	51	BED0	023B	508	40\$:			
		05	023B	509	POPL	R1		
			023E	510	RSB			
			023F	511				

```
023F 513 .SBTTL SEND COMMAND TO CONSOLE
023F 514
023F 515
023F 516
023F 517
023F 518
023F 519
023F 520
023F 521
023F 522
023F 523
023F 524
023F 525
023F 526
023F 527
023F 528
023F 529
023F 530
023F 531
023F 532
023F 533
023F 534
023F 535
023F 536
023F 537
023F 538
023F 539
023F 540
0244 541
0247 542
0248 543
024E 544
0250 545
0253 546
0256 547
025A 548
025B 549
025B 550
025E 551

      CON$SENDCONSCMD - SEND CPU-DEPENDENT COMMAND TO CONSOLE
      FUNCTIONAL DESCRIPTION:
      INITIATE FUNCTION ON CONSOLE
      INPUTS:
      R0 = CONSOLE FUNCTION TO PERFORM:
          CON$C_BOOTCPU = SEND REBOOT SIGNAL TO CONSOLE AND THEN HALT
          CON$C_CLRWARM = CLEAR CONSOLE WARMSTART FLAG
          CON$C_CLRCOLD = CLEAR CONSOLE COLDSTART FLAG
      R2 = NUMBER OF BYTES OF DATA TO BE RETURNED (= 0 IF NO DATA EXPECTED)
          (CURRENTLY ONLY IMPLEMENTED IN 11/790 VERSION OF THIS ROUTINE)
      R3 = ADDRESS OF BUFFER TO HOLD RETURNED DATA (ONLY IF R2 IS NON-ZERO)
          (CURRENTLY IMPLEMENTED ONLY IN 11/790 VERSION OF THIS ROUTINE)
      OUTPUTS:
      CONSOLE STATE MODIFIED
      R1 DESTROYED
      --
CON$SENDCONSCMD::
      BISW    #^XFOO,R0          ; SELECT MISCELLANEOUS CONSOLE COMM.
      MFPR    #PR$ TXCS,R1       ; GET TRANSMITTER STATUS
      BBC     #7,RT,10$          ; WAIT FOR CONSOLE READY
      CMPB    R0,#CON$C_BOOTCPU  ; REBOOT CPU?
      BEQL    30$                ; IF SO BRANCH TO HALT AFTER COMMAND
      MTPR    R0,#PR$ TXDB       ; OTHERWISE ASSERT COMMAND
      MFPR    #PR$ TXCS,R1       ; GET TRANSMITTER STATUS
      BBC     #7,RT,20$          ; WAIT FOR CONSOLE DONE
      RSB     ; RETURN
      10$:    MTPR    R0,#PR$ TXDB ; SEND REBOOT COMMAND TO CONSOLE
      20$:    MTPR    R0,#PR$ TXDB
      30$:    MTPR    R0,#PR$ TXDB
      HALT
```

```
025F 553 .SBTTL "ALLOCATE" CONSOLE TERMINAL
025F 554
025F 555 :++
025F 556 :CONSOWNCTY - "ALLOCATE" CONSOLE TERMINAL
025F 557 :
025F 558 :FUNCTIONAL DESCRIPTION:
025F 559 :
025F 560 :THIS ROUTINE SHOULD BE CALLED WHEN PERFORMING NON-INTERRUPT DRIVEN
025F 561 :I/O TO THE CONSOLE TERMINAL. IT DISABLES INTERRUPTS AND DOES ANY
025F 562 :CPU-SPECIFIC INITIALIZATION OF THE CONSOLE TERMINAL REGISTERS.
025F 563 :CONSRELEASECTY SHOULD BE CALLED TO RESTORE THE STATE OF THE CONSOLE
025F 564 :TERMINAL INTERFACE REGISTERS.
025F 565 :
025F 566 :INPUTS:
025F 567 :NONE
025F 568 :
025F 569 :OUTPUTS:
025F 570 :
025F 571 :R0: VALUE TO BE RESTORED TO TXCS WHEN RELEASING CONSOLE TTY
025F 572 :R1: VALUE TO BE RESTORED TO RXCS WHEN RELEASING CONSOLE TTY
025F 573 :
025F 574 :PR$ RXCS AND PR$ TXCS ARE SET UP SO THAT NON-INTERRUPT I/O CAN BE
025F 575 :PERFORMED TO THE CONSOLE TERMINAL.
025F 576 :
025F 577 :11/780, 11/750, AND 11/730:
025F 578 :CONSOLE INTERRUPTS ARE DISABLED
025F 579 :
025F 580 :11/790:
025F 581 :CONSOLE TRANSMIT AND RECEIVE MASKS ARE SET UP SO THAT ONLY
025F 582 :I/O TO THE CONSOLE TERMINAL IS PERMITTED. INTERRUPTS ARE
025F 583 :DISABLED.
025F 584 :--
025F 585 :CONSOWNCTY::
50 22 DB 025F 586 MFPR #PR$ TXCS,R0 : GET VALUE TO BE RESTORED TO TXCS.
51 20 DB 0262 587 MFPR #PR$ RXCS,R1 : GET VALUE TO BE RESTORED TO RXCS.
20 00 DA 0265 588 MTPR #0,#PR$ RXCS : DISABLE RECEIVE INTERRUPTS
22 00 DA 0268 589 MTPR #0,#PR$ TXCS : DISABLE TRANSMIT INTERRUPTS
05 05 026B 590 RSB
```



```
026C 592 .SBTTL RELEASE CONSOLE TERMINAL
026C 593
026C 594 ++
026C 595 :CONSRELEASECTY - RELEASE CONSOLE TERMINAL
026C 596
026C 597 FUNCTIONAL DESCRIPTION:
026C 598
026C 599 THIS ROUTINE SHOULD BE CALLED TO RELINQUISH EXCLUSIVE USE OF THE
026C 600 CONSOLE TERMINAL OBTAINED BY CALLING CONSOWNCTY. IT RESTORES THE
026C 601 STATE OF THE CONSOLE REGISTERS RXCS AND TXCS.
026C 602
026C 603 INPUTS:
026C 604 R0: VALUE RETURNED BY CONSOWNCTY TO BE RESTORED TO TXCS
026C 605 R1: VALUE RETURNED BY CONSOWNCTY TO BE RESTORED TO RXCS
026C 606
026C 607 OUTPUTS:
026C 608 RXCS AND TXCS ARE RESTORED TO THEIR ORIGINAL STATE.
026C 609
026C 610 --
026C 611 CONSRELEASECTY::
22 50 DA 026C 612 MTPR R0,#PRS_TXCS ; RESTORE TXCS
20 51 DA 026C 613 MTPR R1,#PRS_RXCS ; RESTORE RXCS
05 0272 614 RSB
0273 615 ++
0273 616 :CONSGETCHAR - GET A CHARACTER FROM THE CONSOLE TERMINAL
0273 617
0273 618 FUNCTIONAL DESCRIPTION:
0273 619
0273 620 THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
0273 621 DIRECTLY TO THE CONSOLE TERMINAL
0273 622
0273 623 INPUTS:
0273 624 None
0273 625
0273 626 OUTPUTS:
0273 627 R0 contains the character.
0273 628
0273 629 --
00000013 0273 630 control_s = 19 ; control s (xoff)
00000011 0273 631 control_q = 17 ; control q (xon)
0273 632
0273 633 CONSGETCHAR::
F9 50 20 DB 0273 634 10$: mfpr #pr$_rxcs,r0 ;receiver ready?
50 07 E1 0276 635 bbc #7,r0,10$ ;if clr, receiver not ready
50 21 DB 027A 636 mfpr #pr$_rxdb,r0 ;read input character
05 027D 637 rsb ;return
027E 638 ++
027E 639 :CONSPUTCHAR - PUT A CHARACTER TO THE CONSOLE TERMINAL
027E 640
027E 641 FUNCTIONAL DESCRIPTION:
027E 642
027E 643 THIS ROUTINE SHOULD BE CALLED TO DO NON-INTERRUPT DRIVEN I/O
027E 644 DIRECTLY TO THE CONSOLE TERMINAL
027E 645
027E 646 INPUTS:
027E 647 R0 - Character to be output
027E 648
```

```
027E 649 : OUTPUTS:
027E 650 : Character written to the console terminal.
027E 651 :
027E 652 :--
027E 653 CON$PUTCHAR::
13 51 07 00 ED 028A 658 cmpzv #0,#7,r1,#control_s ;control-s?
11 51 07 00 EF 12 028F 659 bneq 30$ ;if neq no
51 20 DB 0291 660 20$: mfpr #pr$ rxcs,r1 ;receiver ready?
F9 51 07 E1 0294 661 bbc #7,rT,30$ ;if clr, receiver not ready
51 21 DB 0298 662 mfpr #pr$ rxdB,r1 ;read input character
11 51 07 00 ED 029B 663 cmpzv #0,#7,r1,#control_q ;is it a control-q?
51 22 DB 02A0 664 bneq 20$ ;no, wait for another character.
F9 51 07 E1 02A5 665 30$: mfpr #pr$ txcs,r1 ;transmitter done?
23 50 DA 02A9 666 bbc #7,rT,30$ ;if clr, transmitter not done
51 BED0 02AC 667 mtp r0,#pr$_txdb ;write output character
05 02AF 668 popl r1 ;restore a register
0280 669 rsb ;return
0280 670
0280 671 CON_END:
0280 672 .END
```

OPDRIVER  
Symbol table

- VAX/VMS CONSOLE TERMINAL DRIVER F 7

16-SEP-1984 00:16:57 VAX/VMS Macro V04-00  
5-SEP-1984 04:11:02 [SYSLOA.SRC]OPDRIVER.MAR;1

Page 20  
(12)

BURST	= 0000018C	R	02	TTY\$M_TANK_PREMPT	= 00000100
CLASS_DDT	= 00000010			TTY\$M_TANK_STOP	= 00000200
CLASS_GETNXT	= 00000000			TTY\$V_PC_NOTIME	= 00000000
CLASS_POWERFAIL	= 00000020			TTY\$V_TANK_BURST	= 0000000B
CLASS_PUTNXT	= 00000004			UCB\$B_TT_DEPARI	= 000000EC
CLASS_SETUP_UCB	= 00000008			UCB\$B_TT_PARITY	= 000000F8
CON\$ABORT	= 000001E7	RG	02	UCB\$B_TT_PREMPT	= 0000010A
CON\$C_BOOTCPU	= 00000002			UCB\$B_DDB	= 00000028
CON\$DISCONNECT	00000086	RG	02	UCB\$B_DDT	= 00000088
CON\$DS_SET	00000086	RG	02	UCB\$B_DEVCHAR	= 00000038
CON\$GETCHAR	00000273	RG	02	UCB\$B_DUETIM	= 0000006C
CON\$INITIAL	00000004	RG	02	UCB\$B_FPC	= 0000000C
CON\$INITLINE	00000017	RG	02	UCB\$B_FR4	= 00000014
CON\$INIT CTY	00000086	RG	02	UCB\$B_TT_CLASS	= 00000114
CON\$INTINP	00000087	RG	02	UCB\$B_TT_GETNXT	= 0000010C
CON\$INTOUT	00000109	RG	02	UCB\$B_TT_OUTADR	= 0000011C
CON\$NULL	00000086	RG	02	UCB\$B_TT_PORT	= 00000118
CON\$OWNCTY	0000025F	RG	02	UCB\$B_TT_PUTNXT	= 00000110
CON\$PUTCHAR	0000027E	RG	02	UCB\$M_INT	= 00000002
CON\$RELEASECTY	0000026C	RG	02	UCB\$M_INTTYPE	= 00000080
CON\$RESUME	00000201	RG	02	UCB\$M_TIM	= 00000001
CON\$SENDCONSCMD	0000023F	RG	02	UCB\$V_INT	= 00000001
CON\$SET_LINE	00000086	RG	02	UCB\$V_INTTYPE	= 00000007
CON\$SET MODEM	00000086	RG	02	UCB\$V_POWER	= 00000005
CON\$STARTIO	000000C0	RG	02	UCB\$V_TIM	= 00000000
CON\$STARTIO1	000000C7	R	02	UCB\$W_STS	= 00000064
CON\$STOP	000001DF	RG	02	UCB\$W_TT_HOLD	= 00000108
CON\$XOFF	000001C1	RG	02	UCB\$W_TT_OUTLEN	= 00000120
CON\$XON	000001C1	RG	02	UCB\$W_TT_PRTCTL	= 00000122
CONTROL_Q	= 00000011			UCB\$W_UNIT	= 00000054
CONTROL_S	= 00000013				
CON_END	000002B0	R	02		
CURR	00000000	R	02		
DATA	00000002	R	02		
DDB\$B_DDT	= 0000000C				
DEV\$V_TRM	= 00000002				
DISMIS	00000152	R	02		
DPT\$W_VECTOR	= 0000001E				
EXE\$GL_ABSTIM	*****	X	02		
EXIT	000001C0	R	02		
EXIT_INT	000001BC	R	02		
FLOPINT	00000138	R	02		
IDB\$B_UCBLST	= 00000018				
IDB\$W_UNITS	= 0000000C				
NEXT	00000001	R	02		
OPAS\$VECTOR	*****	X	02		
PR\$_RXCS	= 00000020				
PR\$_RXDB	= 00000021				
PR\$_TXCS	= 00000022				
PR\$_TXDB	= 00000023				
PREMPT	000001AD	R	02		
START_BURST	00000185	R	02		
STOP	000001A7	R	02		
TERMINALIO	0000015C	R	02		
TTY\$GB_PARITY	*****	X	02		
TTY\$GL_DPT	*****	X	02		
TTY\$M_TANK_BURST	= 00000800				



+-----+  
! Psect synopsis !  
+-----+

PSECT name	Allocation	PSECT No.	Attributes
ABS	00000000 ( 0.)	00 ( 0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$ABSS	00000000 ( 0.)	01 ( 1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
SYSLOA	00000280 ( 688.)	02 ( 2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

+-----+  
! Performance indicators !  
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.51
Command processing	146	00:00:00.46	00:00:02.29
Pass 1	490	00:00:12.31	00:00:51.07
Symbol table sort	0	00:00:01.95	00:00:05.34
Pass 2	127	00:00:02.41	00:00:08.62
Symbol table output	10	00:00:00.07	00:00:00.20
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	806	00:00:17.27	00:01:09.05

The working set limit was 1800 pages.

101758 bytes (199 pages) of virtual memory were used to buffer the intermediate code.

There were 100 pages of symbol table space allocated to hold 1915 non-local and 31 local symbols.

672 source lines were read in Pass 1, producing 15 object records in Pass 2.

51 pages of virtual memory were used to define 48 macros.

+-----+  
! Macro library statistics !  
+-----+

Macro library name	Macros defined
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	21
\$255\$DUA28:[SYS.LIB]STARLET.MLB;2	7
TOTALS (all libraries)	28

2291 GETS were required to define 28 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:OPDRIVER/OBJ=OBJ\$:OPDRIVER MSRC\$:OPDRIVER/UPDATE=(ENH\$:OPDRIVER)+EXECMLS/LIB



0398

AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY